Transmitter Classification With Supervised Deep Learning

Cyrille Morin, Leonardo Cardoso, Jakob Hoydis, Jean-Marie Gorce, Thibaud Vial

▶ **To cite this version:**

Cyrille Morin, Leonardo Cardoso, Jakob Hoydis, Jean-Marie Gorce, Thibaud Vial. Transmitter Classification With Supervised Deep Learning. CROWNCOM 2019 - 14th EAI International conference on Cognitive Radio Oriented Wireless Networks, Jun 2019, Poznan, Poland. pp.1-14. hal-02132970

HAL Id: hal-02132970
https://inria.hal.science/hal-02132970

Submitted on 20 May 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Transmitter Classification With Supervised Deep Learning

Cyrille Morin[1 [0000−0001−5878−3501]],
Leonardo S. Cardoso[2,3 [0000−0002−6647−6031]],
Jakob Hoydis[2,3 [0000−0002−0438−967X]], Jean-Marie Gorce[2,3 [0000−0002−5389−0102]],
and Thibaud Vial[3]

[1] Univ Lyon, Inria, INSA Lyon, CITI, France
<first name>.<last name>@inria.fr
[2] Nokia Bell Labs, France
jakob.hoydis@nokia-bell-labs.com
[3] RTone, France
thibaud.vial@rtone.fr

Abstract. Hardware imperfections in RF transmitters introduce features that can be used to identify a specific transmitter amongst others. Supervised deep learning has shown good performance in this task but using datasets not applicable to real world situations where topologies evolve over time. To remedy this, the work rests on a series of datasets gathered in the Future Internet of Things / Cognitive Radio Testbed [4] (FIT/CorteXlab) to train a convolutional neural network (CNN), where focus has been given to reduce channel bias that has plagued previous works and constrained them to a constant environment or to simulations. The most challenging scenarios provide the trained neural network with resilience and show insight on the best signal type to use for identification, namely packet preamble. The generated datasets are published on the Machine Learning For Communications Emerging Technologies Initiatives web site[4] in the hope that they serve as stepping stones for future progress in the area. The community is also invited to reproduce the studied scenarios and results by generating new datasets in FIT/CorteXlab.

Keywords: Transmitter Identification · RF fingerprinting · Deep Learning.

1 Introduction

Communication systems' constant evolution requires a constant search for new techniques that allow to squeeze out every bit of performance out of the system, a constant need to improve spectral efficiency in order to achieve the theoretical maximum capacity given by Shannon's law. This requirement is all the more

[4] Datasets and usage and generation scripts can also be found there: https://wiki.cortexlab.fr/doku.php?id=tx-id

important in systems that transmit many small packets, like Internet of Things (IoT), where currently headers may outweigh the number of payload bits transmitted. Furthermore, headers are currently the only barrier against transmitter identification errors and transmitter impersonation on edge devices that don't have the ressources to use cryptographic protocols. Indeed, security issues are of utmost importance in this new era where hackers are able to easily attack transmissions even at the physical layer. Hence, a new means to provide secure identification of transmissions is needed, both to improve security as well as to render headers a thing of the past.

In the last five years, supervised deep learning (SDL) has imposed itself as the tool to achieve state-of-the-art performance in many fields, starting with image processing to voice recognition, product suggestion, and more generally data analysis and signal processing in physics, medicine and consumer products. SDL really shines in cases where labelled data is plentiful and mathematical models are not known. In the radio communication world, data is generated by the Terabyte per second all over the world, but for traditional applications, precise models already exist. The existence of those good models explains why SDL is not yet widely used in radio communication. Yet it's starting to gain traction in the last couple of years, especially with channel decoding [6] and spectrum monitoring [7]. Indeed, the tool promises increased performance in areas where models are yet to be derived and algorithms are not yet practical for real time implementations. The task of identifying transmitters has attracted some attention in the last years, with two different approaches:

First, the confirmation of identity, by comparing a received signal with a previously authenticated one to verify if those characteristics match. This approach facilitates handling of changing environment by constantly updating the stored authenticated signal, as long as transmissions are more frequent than channel variations. The fingerprints for each device is not stored inside of the identification system but in the recorded signals. It can allow the identification of emitters unseen at training time, but increases processing times: comparisons need to be made with every known device. In [12], the authors study channel responses in a simulated building floor to emulate spatial variations. In [9], channel response is also used, but this time using a Gaussian Mixture Model to study similarities between samples gathered on real radio devices.

In second come the classifiers, and these are the more recent and numerous works, where a system is tasked not to give a similarity score between current and previous samples but to directly output the identity of the transmitter amongst a pool of previously seen radios. A convolutional neural network (CNN) is used in [10] to estimate IQ imbalance parameters of incoming signal from a simulation environment and these parameters are used in a classical Bayesian decision process. Then [2] leverages parameters estimated routinely by decoding systems: DC and frequency offsets, IQ imbalance and channel information as input to a neural network. Matlab simulations show 99% accuracy with up to 10 000 devices. In [5] the authors use 7 consumer Zigbee transmitters to gather data. They remove the decoded data from the received signal to isolate channel

and transmitter information effects before feeding it to a CNN and achieve up to 90% classification accuracy. However, one could argue that some hardware effects depend on the transmitted signal, for example amplifier non linearities, so removing it could be detrimental to the identification process. Different machine learning techniques are evaluated in [13] over a dataset of signals gathered with real USRP devices accounting for 6 radio interfaces. They develop an architecture called a multi layer perceptron made of a network of small neural networks and introduce the use of wavelet transforms with the goal of learning to classify with as few training samples as possible. The work in [3] focuses on amplifier characteristics and measures non linearity variations between 7 USRP cards. The data is used to train a network on hundreds of simulated devices. It also shows the effects of local oscillator leakage on classification accuracy. Finally, in [8] attention is switched to IQ imbalance and DC offset to perform classification. A CNN is trained on dataset made with 16 different USRP devices but it is not able to cope with changes in environment between experiments. For this reason, artificial impairments are added to the signal and increase classification accuracy. Still in the same context, the work in [1] deals with wired communications to increase security and prevent intrusion of malicious systems in the network inside a car.

As previously stated, some works base the identification of transmitters on characteristics outside of the scope of the transmitter radios themselves using, for example, the channels [12,9]. In realistic applications, however, radios are rarely at fixed locations and channel characteristics evolve with the surrounding environment. A better identification method would be to base the identification on the radio frequency (RF) *signature* of the radios themselves, as addressed in [10,3,2,8]. However, in those works, either simulations or simple datasets were used which may not provide a biased free classifier that actually focuses on RF signatures. Furthermore, in those works, no attention is given on the quality of the dataset itself for the identification task at hand (absence of bias, reproducibility, interference with outside sources), and in [8], artificial impairments remove possible security claims: if identification is done on software added elements, any malicious software can do the same and impersonate the user.

To counter the problems encountered by previous works, herein, an extensive dataset campaign was generated aiming to train a more robust classifier. The datasets were carefully crafted to avoid biases like channel, transmitted power, packet structure, and receiver position through different measurement campaigns in a controlled environment. We use the Future Internet of Things / Cognitive Radio Testbed [4] (FIT/CorteXlab) to gather experimental datasets to train neural network classifiers able to identify emitters based on their hardware characteristics. The datasets are available online, as well as the scripts used to generate them, so that anyone can reproduce them on FIT/CorteXlab.

The remainder of this paper is organised as follows. Section 2 deals with the identification problem itself and the characteristics that make it particular. It also describes the FIT/CorteXlab testbed, used for the measurement campaign. Section 3 describes the dataset generation process required to train a signa-

ture based CNN. Then, in Section 4 the CNN structure is described as well as the training process. Results are presented and discussed in Section 5. Finally, conclusions and perspectives are drawn in Section 6.

2 The Identification Problem

Current authentication schemes for packet transmission are based on transmitting an identification number inside of a frame header. This poses two problems:

- The number needs spectral and energy resources to be transmitted. In IoT protocols, these resources are already very limited.
- The identification is trivial to spoof. Meaning that the transmission needs to be authenticated again at higher levels with cryptographic means, or in cases where computing power or energy is limited it cannot be done.

2.1 Characteristic elements of a point-to-point transmission

Channel effects First and most obvious is the impact of the channel between emitter and receiver. In the standard case of a transmission with a fixed average power, the reception power is a direct indicator of the path loss and thus of the distance between emission and reception. This power can then serve as a coarse indicator of the emitter. In this case, multi-path parameters can also be easily measured. These are highly dependent on the position of the emitter.

These two elements have a high impact on the signal and are simple to measure, but they are dependent on the system topology which is expected to change over time, not on intrinsic properties of the radio cards. This could still be used as a way to detect impersonation by looking at sudden changes in these parameters, as studied in [11,12].

Power amplifier imperfections Homodyne radios suffer from IQ imbalance. The In-phase and Quadrature components of the signal do not go through the same path with the same components. This means that they may not be amplified by the same gain and the resulting constellation gets skewed. The second part of the imbalance comes from the fact that the two parts are not mixed with sine waves at exactly 90 degrees and the resulting constellation gets rotated. This effect is used greatly in [8] and [10] but it's mostly present in devices with a homodyne architecture as those used in software refined radio (SDR) and not in superheterodyne radios that make up the majority of consumer devices.

Amplifiers of RF signal are non linear components. They are setup so they function mostly in their linear region but, even there, their response curve is not perfectly linear. The parameters corresponding to this are slightly different from one radio chip to another, even amongst the same product line [3].

Local oscillator imperfections The local oscillator is tasked with translating the baseband signal to the carrier frequency. It is however not able to set itself to the exact same frequency as the receiver. This creates a frequency offset that may be characteristic of an emitter The offset is not only dependent on intrinsic elements of the radio card, but also on temperature so it can change over time at a rate that can also be indicative of the emitter, if it's high enough to be measured.

The frequency translation operation that uses the oscillator to bring baseband signals to carrier frequency can present a local oscillator leak in homodyne devices. In this case, a peak is sent at the oscillator frequency, whose power does not depend on the actual signal sent, but only on the amplifier settings. The phenomenon is used in [3] to identify emitters even when, considering data sent, $SNR = -\infty$.

2.2 Avoiding dataset bias

The aforementioned elements are not all desirable: The channel effects are only distinctive of the position of the emitter and may not be reliable when the environment becomes realistically dynamic. Some of the other imperfections are specific to homodyne radio cards. This means that they can appear in the USRPs that will be used, but not on many mainstream consumer grade superheterodyne radios. So reducing the reliance on these imperfections allows better generalisation of the results to other radio hardware architectures.

The USRPs are calibrated to remove IQ imbalance and DC offsets and the emission gain is set to reduce local oscillator leakage to a minimum. But channel effects cannot be removed as simply while still maintaining realistic over-the-air radio propagation (no cable). Instead of removing them, they can be randomised: When a parameter is random and varies over a range that overlaps these of the other emitters, one specific realisation of that random parameter cannot give insight on who is the transmitter. In this case, the neural network (NN) will not learn to depend on that parameter to perform identification.

2.3 The FIT/CorteXlab platform for learning

The FIT/CorteXlab testbed is a SDR testing facility in France that enables testing of physical (and higher) layer techniques for future wireless systems. It counts with 42-node high-performance SDR nodes, whose frequency range is roughly from 400 MHz to 4 GHz, at 20 MHz of maximum bandwidth. It is fully accessible to the wireless research community and uses GNU Radio as its programming environment. One of its main characteristics, the one that makes FIT/CorteXlab particularly well suited for machine learning (ML), is its shielded room. It allows for reproducible experiments since the shielding provides a fully a controlled environment. Also important for ML is the presence of a server equipped with GPUs connected via high speed data links to the SDR nodes, which allow on-the-fly training and exploitation of SDL techniques.

3 Dataset generation process

The data generation process is modular to allow for testing of a wide range of scenarios with various amount of bias. There is a standard core process that can be parameterised to create different scenarios. The signal processing chain is written as GNU Radio flowgraphs for emission and reception that provide a light API for configuration. All the high-level management of the experiments is done in small python and bash scripts for easy configuration and reproduction of scenarios.

3.1 Core process

There are 21 emitters, one receiver, and one scheduler. Each of these, except the scheduler, uses a National Instruments USRP N2932 SDR working at 5 Msample/s and 433 MHz. All the emitters have to transmit packets to the receiver and use the same frequency band. The scheduler's role is to ensure that there is no interference between packets from different emitters without needing to implement carrier sensing algorithm. It also causes packets from different emitters to be sent in close temporal proximity. This is useful if the environment is not static: one specific environment configuration could be indicative of a specific emitter if it was not the case. Every millisecond the scheduler selects randomly one among the possible emitters, then sends a packet to it via UDP.

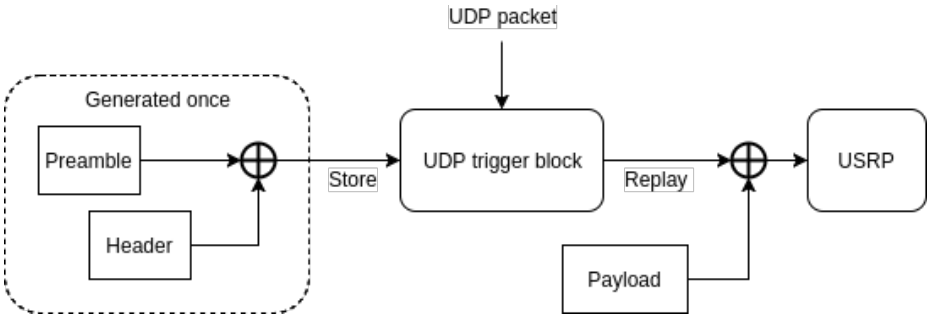

Fig. 1. Simplified emitter flowgraph

The emitters have their USRP set to burst mode. This means that when they are not actively transmitting, their amplifiers are off so they do not emit anything. Even local oscillator leakage is prevented by this. Upon reception of a scheduler packet, an emitter wakes up its USRP, waits for the amplifier to stabilise and sends a frame. The frame is composed of three parts:

- A known preamble for detection and time synchronisation with the frame.
- A header: an OFDM frame containing the identification of the emitter.

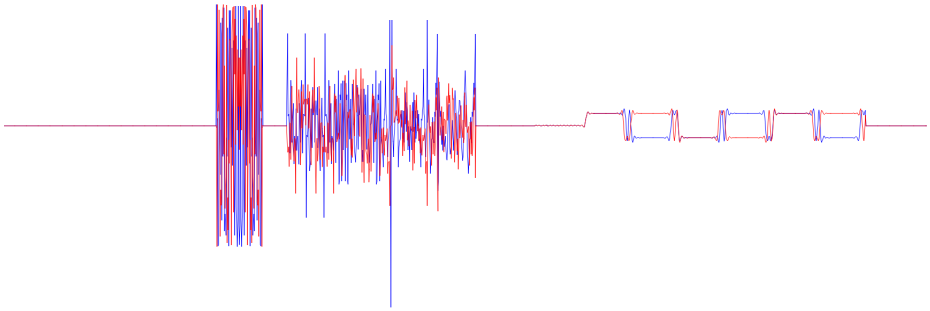

Fig. 2. Frame samples sent to USRP for emission, with zeroes for amplifier wake up, preamble, header, and payload with guard intervals

– The payload that we are interested in, containing either noise, a random or a static QPSK modulated sequence of 560 samples long. In all these cases, the payload does not contain any emitter specific information.

There is a time gap between the header and the payload to give the amplifier time to stabilise after the header and avoid interference between the two parts if there is a significant amplitude difference between the two parts.

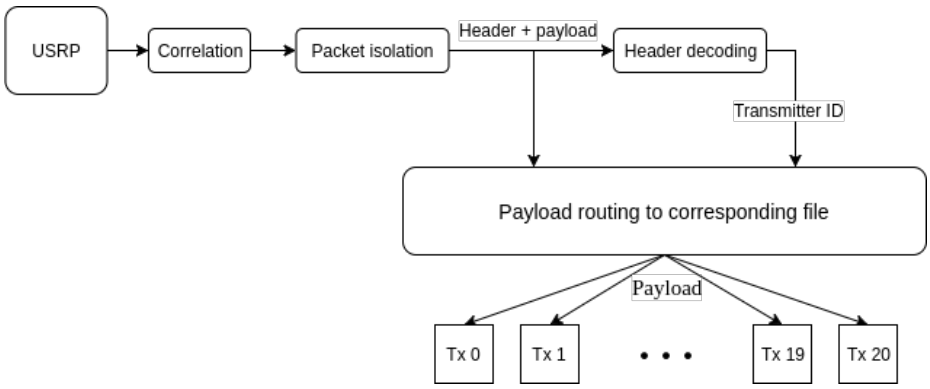

Fig. 3. Reception flowgraph

The receiver's USRP stays on for the duration of the experiment. It uses a correlator to detect the presence and the timing of the known preamble, then isolates the number of samples corresponding to the length of the header and

the preamble. An OFDM packet receiver is used to decode the header and the identification number is used to send the payload samples to be recorded in the corresponding file.

The grouping of the recorded files for one experiment in one scenario forms a dataset. Experiment run times are set to gather about 50000 packets per emitter.

3.2 Scenarios

An experiment scenario has two main parameters to allow testing and selection of different elements:

Type of signal The goal here is to identify the emitter based on how it sends data, not what it sends. But every type of data does not necessarily yield the same classification accuracy. To show this, three types of payload are tested:

- A fixed sequence of QPSK modulated bits: All the emitters always send the exact same sequence: the bit sequence of the 802.15.4 preamble. This reduces the noise the CNN has to deal with while still being a realistic case: here it's not the user data that is used but the preamble and this has to be transmitted anyways.
- A random sequence of modulated bits: The emitters generate random sequences of bits and these are modulated in the same way for every emitter. QPSK is used as it is a commonly used modulation scheme for IoT devices.
- A noise sequence: The payloads are randomly uniformly generated from a noise source. This allows to test if the modulation choice has an impact on the performance or if any modulation would work.

Transmission complexity

- Plain: The simplest of the modes. All the payloads from all emitters are sent with the same amplitude and nothing moves inside of the experimentation room. This is mostly used as a benchmark to compare the other channel randomising scenarios.
- Varying amplitude: The emission amplitude varies from one payload to another to emulate changes in path loss for each emitter without physically moving them. This is implemented by scaling the IQ samples before sending them to the USRP, and not by changing the gain settings of the device because the amplifier takes a relatively long time to stabilise whereas scaling samples in software is instantaneous. Nothing moves inside of the room, so the multipath parameters are still static.
- Robot: The payloads are sent the same way as in the previous scenario. A robot is introduced, covered with metallic sheets to increase radio waves reflections and set to move randomly inside the room. This introduces new and constantly changing reflections to randomise the multipath parameters.

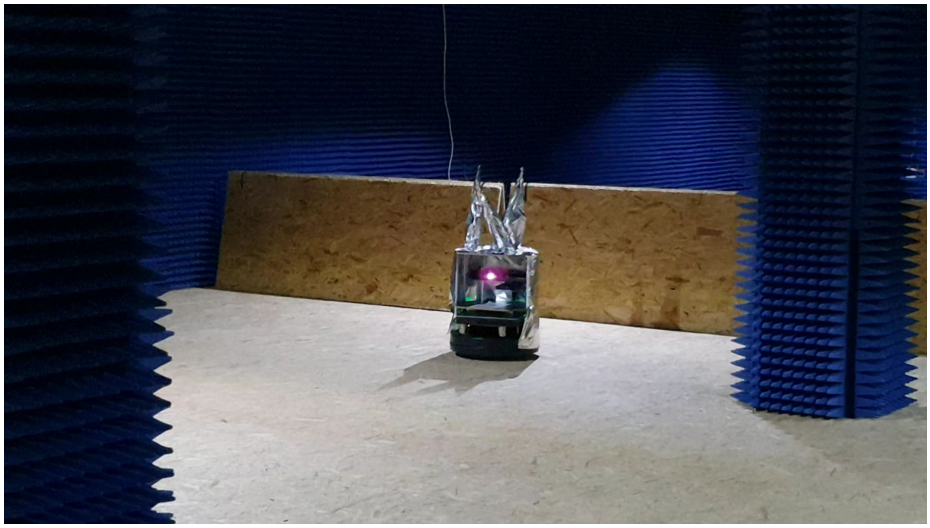

Fig. 4. The Turtlebot robot with the metallic sheets, inside of FIT/CorteXlab

4 Learn to classify

4.1 System architecture

We use a CNN type network with five layers of convolution and six dense layers. It takes 600 complex samples organised in a matrix of 600x2 float numbers for the Cartesian coordinates of the complex numbers. And it outputs a vector of 21 numbers corresponding to the likelihood that the input was from one of the 21 transmitters. Each layer has an exponential linear unit (ELU) activation function except for the output layer which uses a softmax activation.

4.2 Training phase

Before training, datasets are randomly shuffled and then split into 3 parts: 70% used for training, 10% for validation and hyperparameter tuning and the last 20% for testing. Networks are trained over all the gathered datasets with the same architecture. Training is done on mini-batches of 128 examples over more than 30 epochs, with an Adam optimiser, 0.001 of learning rate, l1 regularisation on the dense layers while minimising the categorical crossentropy loss function.

Hyperparameter tuning was done by training networks with increasing amount of dense and convolutional layers, with varying batch sizes, learning rates and regularisation on a dataset that was found to be hard to train on: Varying amplitude and a payload of random bits. They were trained for ten epochs and the best performing was selected.

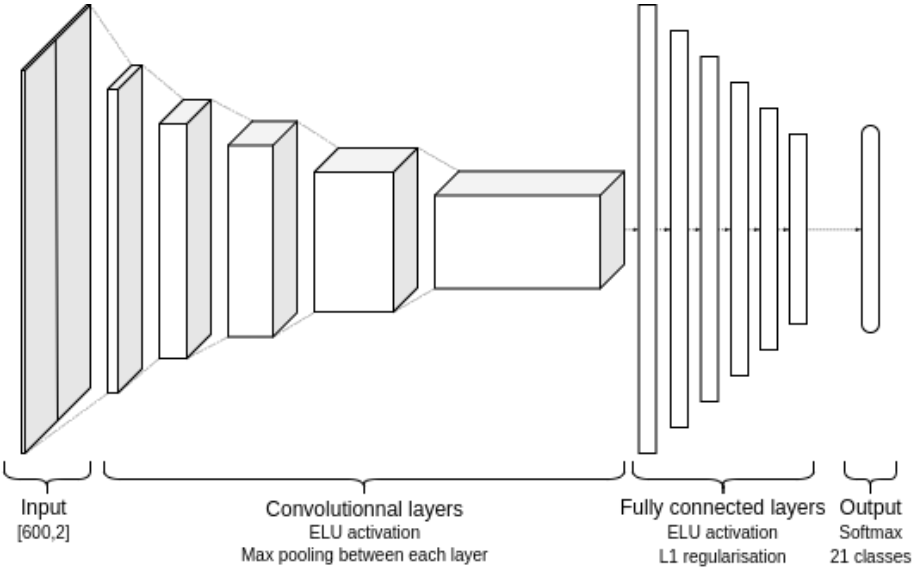

Fig. 5. Neural network architecture

5 Results

The first step is to establish a benchmark with the simplest scenario. This actually shows a comparison with the network in [8] without artificial impairments. In [8], they use a very similar setup of 16 static USRPs placed in a room with no moving object. Their network achieves a classification accuracy of 98.6 % whereas the one studied here achieves 99.9 % with more classes (21 instead of 16)

Fig. 6 presents the classification accuracy achieved for the three signal types, while comparing the Plain and Varying amplitude scenarios. Having randomness in the transmitted signal degrades classification accuracy and this degradation increases with the scenario complexity. However, we can also see that having a completely random noise does not cause a significant performance loss over a QPSK modulated signal, even though the latter has a limited constellation size compared to noise. Thus a higher order modulation should not cause further accuracy losses, but the ideal case is to use a static signal, as would be a frame preamble.

Fig. 7 studies the impact of environment modification on classification accuracy for the three scenarios. In it, one can observe that, the more complex a scenario is, the harder it is for a network to train on it, by a slight margin but the better it is at resisting changes in the environment. The training datasets were generated one after another, with no changes inside the FIT/CorteXlab room, then a metallic stool was introduced inside the room and the other datasets were

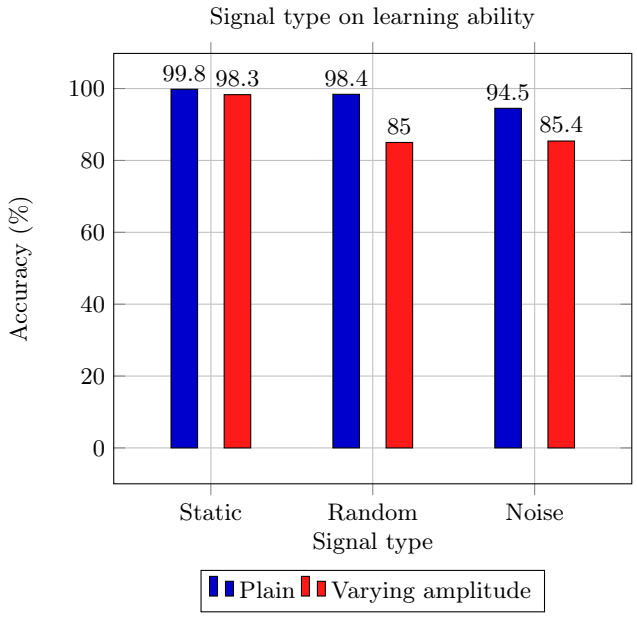

Fig. 6. Accuracy reached by networks trained on plain or varying amplitude scenarios and with various signal types. The accuracy is measured on the test set from the training dataset.

generated. This ensures that the change in signal propagation is exactly the same for the 3 scenarios. The Plain scenario suffers from a big loss of accuracy, as was noticed in [8], but the Robot one is very resilient to this kind of change.

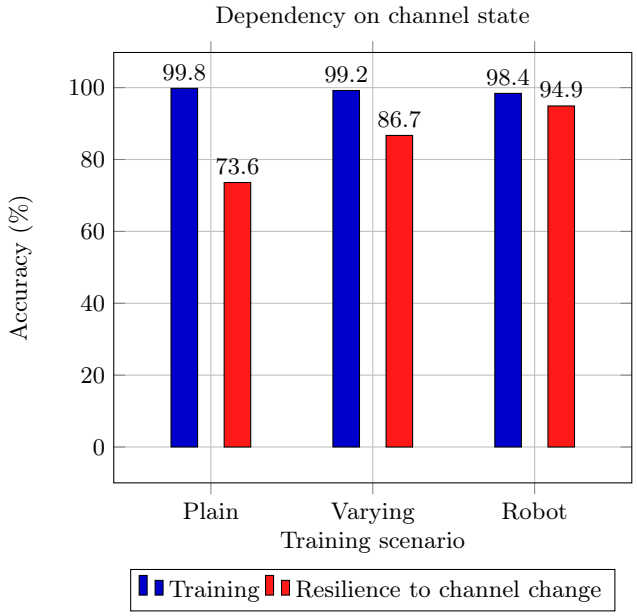

Fig. 7. Accuracy of networks trained one one scenario with static signal and tested, either on test data from the training dataset or on a dataset with the same scenario but with a chair added to the room.

Finally, Fig.8 focuses on the ability to generalise to other scenario types. A network is trained on each scenario and tested on all of them. We can observe a decrease in accuracy when a simple scenario is tested on a more complex one. From this, one can infer that the more random the channel is, the more the trained network is able to cope with a change of scenario and also with a change in the environment.

The key takeaway from these results is as follows: if one were to implement a transmitter identification system in a production setting, its goal should be to train the neural network with the maximum amount of channel variability.

6 Conclusions

The task of identifying transmitters based on hardware physical characteristics and imperfections has been gathering attention in the literature recently. These identification strategies have been based aspects such as IQ imbalance, amplifier non linearities or channel properties. Most of the works in the area involve the use

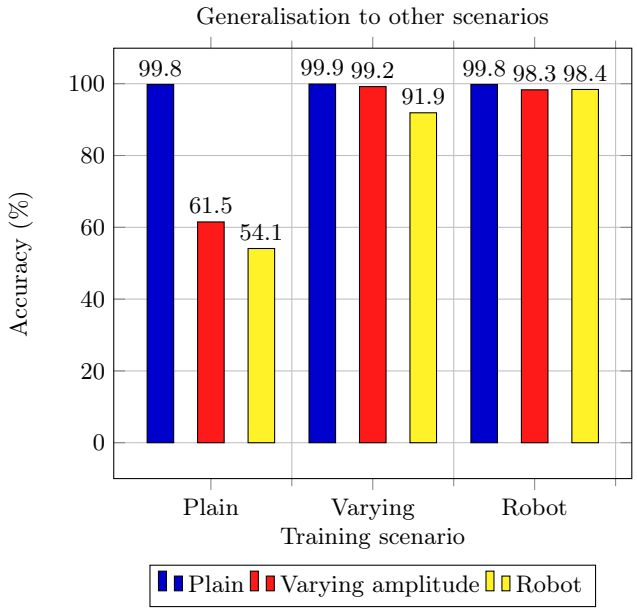

Fig. 8. Accuracy of networks trained on one scenario and tested on the others. Here the signal is of the static type and the environment is untouched.

of machine learning, and more specifically deep learning with good identification accuracies. However, the data used to train the neural networks from prior works can not guarantee bias avoidance against unwanted elements, such as channel effects.

In this work, instead of focusing on a specific radio imperfection, neural networks were trained on raw IQ samples so as not to overlook any effect. On the other hand, data was generated with the goal of minimising the role of the unwanted channel on the identification task by randomising the various channel parameters.

The considered neural network architecture shows state-of-the-art performance when tested on similar scenarios as previous works. An exploration of various parameters was done with results that shows that the identification task is simpler when transmitted signals do not change and that unknown signals only incur a small performance decrease, independently of the modulation type. They also show that training data with increased complexity do not impede learning ability but provides increased robustness against environment modifications. Finally, from the experiments and the results, we see that FIT/CorteXlab provides the stability and reproducibility necessary for machine learning approaches.

We plan to extend this setup to use packets recorded from not just one but from many receivers to increase the randomness of the perceived channel with the transmitters and evaluate the robustness of this approach to environment modification. Another direction of exploration is to verify that higher order

modulation schemes do not impair classification accuracy with respect to what observed with QPSK.

Acknowledgement

This work was supported by Inria Nokia Bell Labs ADR "Analytics and machine learning for mobile networks". Experiments presented in this paper were carried out using the FIT/CorteXlab testbed. (see http://www.cortexlab.fr).

References

1. Avatefipour, O., Hafeez, A., Tayyab, M., Malik, H.: Linking received packet to the transmitter through physical-fingerprinting of controller area network (jan 2018), http://arxiv.org/abs/1801.09011
2. Chatterjee, B., Das, D., Sen, S.: RF-PUF: IoT security enhancement through authentication of wireless nodes using in-situ machine learning (may 2018), http://arxiv.org/abs/1805.01048
3. Hanna, S.S., Cabric, D.: Deep learning based transmitter identification using power amplifier nonlinearity (nov 2018), http://arxiv.org/abs/1811.04521
4. Massouri, A., Cardoso, L., Guillon, B., Hutu, F., Villemaud, G., Risset, T., Gorce, J.M.: Cortexlab: An open FPGA-based facility for testing SDR & cognitive radio networks in a reproducible environment. In: Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on. pp. 103–104. IEEE (2014). https://doi.org/10.1109/INFCOMW.2014.6849176
5. Merchant, K., Revay, S., Stantchev, G., Nousain, B.: Deep learning for RF device fingerprinting in cognitive communication networks. IEEE Journal of Selected Topics in Signal Processing **12**(1), 160–167 (feb 2018). https://doi.org/10.1109/JSTSP.2018.2796446
6. Nachmani, E., Beery, Y., Burshtein, D.: Learning to decode linear codes using deep learning (jul 2016), http://arxiv.org/abs/1607.04793
7. O'Shea, T.J., Clancy, T.C., McGwier, R.W.: Recurrent neural radio anomaly detection (nov 2016), http://arxiv.org/abs/1611.00301
8. Sankhe, K., Belgiovine, M., Zhou, F., Riyaz, S., Ioannidis, S., Chowdhury, K.: ORACLE: Optimized Radio clAssification through Convolutional neuraL nEtworks (dec 2018), http://arxiv.org/abs/1812.01124
9. Weinand, A., Karrenbauer, M., Sattiraju, R., Schotten, H.D.: Application of machine learning for channel based message authentication in mission critical machine type communication (nov 2017), http://arxiv.org/abs/1711.05088
10. Wong, L.J., Headley, W.C., Michaels, A.J.: Emitter identification using CNN IQ imbalance estimators (aug 2018), http://arxiv.org/abs/1808.02369
11. Xiao, L., Greenstein, L., Mandayam, N., Trappe, W.: Fingerprints in the ether: using the physical layer for wireless authentication (jul 2009), http://arxiv.org/abs/0907.4877
12. Xiao, L., Greenstein, L., Mandayam, N., Trappe, W.: Using the physical layer for wireless authentication in time-variant channels (jul 2009). https://doi.org/10.1109/TWC.2008.070194
13. Youssef, K., Bouchard, L.S., Haigh, K.Z., Krovi, H., Silovsky, J., Valk, C.P.V.: Machine learning approach to RF transmitter identification (nov 2017), http://arxiv.org/abs/1711.01559